

Tarea grande de C: Torneo de CARNAC

PROF. KIRSTEIN GÄTJENS S.

**"El patriotismo en el campo de batalla consiste en conseguir
que otro desgraciado muera por su país antes de que consiga
que tu mueras por el tuyo."**

- George S. Patton

Instrucciones generales:

- + Entregue los fuentes en alto nivel necesarios para recompilar su tarea.
- + Debe entregarse antes del lunes 17 de abril antes de la medianoche al correo kirstein.eval@gmail.com
- + El nombre del programa principal debe ser `carnac-Apellido-nombre.c`
- + Los módulos se pueden llamar como `deseen`
- + El subject del correo es lo usual: TAREA: Carnac
- + El cuerpo del correo debe contener su nombre completo, número de carne y curso.
- + La documentación a entregar debe ser un PDF aparte, debido a que es una tarea grande. Es importante además de lo que se solicita siempre en las tareas del curso, incluir documentación sobre la IA programada y el algoritmo de aleatoriedad.
- + Entregue un compreso con todas las cosas. El nombre del compreso debe ser: `Carnac-Apellido-Nombre.RAR`
- + Siempre es indispensable que el código y los datos posean abundantes y oportunos comentarios.
- + Esta tarea grande vale por 15 resúmenes numerados del xx al zz.
- + Es una tarea grande con el objetivo de que solo hagamos una tarea de C y nos divirtamos mucho. Eso sin contar con lo divertido de programar una IA que juegue aceptablemente.

La tarea consiste en realizar una pequeña versión del juego CARNAC.

La idea será jugar un torneo de este juego por un pote de 50 puntos del proyecto.
El proyecto base sin torneo valdrá 80%.

Para tener derecho a jugar el torneo el juego debe cumplir con los requisitos de torneo, marcados en este enunciado como RdT.

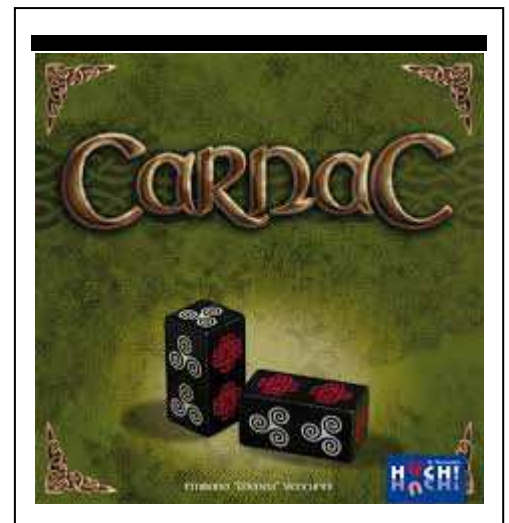
La tarea base incluye el jugar legalmente e inteligentemente. La tarea base incluye la interfaz completa. No toda la tarea base es requisito para participar en el torneo.

El juego debe programarse en C, teniendo como opciones el CC de linux o el TC 2.0 No pueden usar otras versiones de dichos softwares.

Deben programar una interfaz gráfica para que se pueda jugar una partida de Carnac.
Debe seguir las reglas del juego oficiales del curso (hay un par de variantes al respecto de los videos y reglas oficiales del juego).

Siempre comenzará quien vaya con las blancas. RdT. (es una regla original nuestra, no del Carnac).
En cada jugada debe poderse seleccionar quien piensa la jugada (persona o IA) RdT.

Se espera que la IA solo pueda jugar con la información del tablero actual.
La IA debe ser lo mejor posible y eso tendrá una nota de la tarea base.



El jugar inteligentemente no es RdT, pero el jugar legalmente sí.

Deben ir desplegando un score que lleve la cuenta de los dolmens actuales en ese momento.

Una vez que la partida termina debe decir quien es el que ganó y porqué.

En caso de extremo empate (misma cantidad de dolmos y todos en tamaños equivalentes) se considerará ganador el rojo (es una regla original nuestra, no del Carnac).

Como siempre se espera un acerca de y una ayuda, pero al ser una tarea gráfica deben ser deslumbrantes.

Salir decentemente del juego es un requisito de la tarea.

Debe en cada jugada irse escribiendo y cerrando un archivo de texto con la bitácora.

La bitácora indica la secuencia de movimientos que se han ido realizando.

Se debe crear con el nombre Bitac000.txt o Bitac001.txt o Bitac002.txt, etc. De forma que no le caigan encima a ningún archivo.

El estándar de las jugadas debe seguirse al pie de la letra. RdT.

Las coordenadas de cada casilla se marcarán: filas con letras y columnas con números comenzando desde 0. Se numeran de arriba hacia abajo y de izquierda a derecha.

La casilla superior izquierda será entonces la casilla A0.

La casilla inferior derecha será entonces la casilla i13

El tablero mediano comenzará siempre en la casilla B2 y el pequeño en la C3.

El tablero mediano terminará en la casilla h11 y el pequeño en la g10.

Para indicar un movimiento se debe decir el color de arriba, la casilla y el color del norte.

Por ejemplo la primera jugada del video de partida de ejemplo:

<https://www.youtube.com/watch?v=H7SHZb97BAM>

sería: E6RB

El contrincante debe decidir si lo vuelca o no lo vuelca y si lo vuelca el sentido de la caída: NSEO

Esto también debe ser posible que lo decida la IA o se haga manualmente. RdT.

En este caso el contrincante jugó a volcar al Sur.

la segunda jugada sería: i6BR y el contrincante decidió volcar al Oeste.

Es importante que el juego diga claramente las decisiones que toma en cada jugada que debe pensar la IA. RdT.

Como se le dice a la interfaz la jugada manual y como muestra la jugada de la IA la interfaz es decisión de diseño y coma tal será calificada, pero el RdT es que se entienda claramente (que no haya ninguna ambigüedad).

Se les recomienda hacerle algo adicional al random que provee el lenguaje de alto nivel.

Se considerará un tiempo prudencial para decidir si la IA está pensando o si ya murió, a criterio del profesor.

Quedarse enciclado no es jugar legalmente.

No es aceptable la técnica de los 28 pasos.

Al inicio de cada partida debe indicarse en que territorio ocurrirá la batalla: Siberia, Alemania o el Vaticano. RdT

¡Suerte!